



Introducing Behavior in Function Blocks

Xavier Rebeuf

► To cite this version:

Xavier Rebeuf. Introducing Behavior in Function Blocks. Séminaire Zhejiang University, 2002, Zhejiang University, Zhejiang/China. inria-00099453

HAL Id: inria-00099453

<https://hal.inria.fr/inria-00099453>

Submitted on 26 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Introducing Behavior in Function Blocks

X. Rebeuf

(Xavier.Rebeuf@loria.fr)

ZheJiang University 2002

Introduction

Distributed industrial-
process control



Compose hardware and
software components

PB: Interoperability between components

Introduction

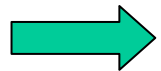
Distributed industrial-
process control



Compose hardware and
software components

PB: Interoperability between components

- Abstraction of elementary component = Function block
- Composition of function blocks = Functional Requirement Diagram



Structural architecture

Introduction

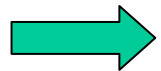
Distributed industrial-
process control



Compose hardware and
software components

PB: Interoperability between components

- Abstraction of elementary component = Function block
- Composition of function blocks = Functional Requirement Diagram



Structural architecture



Validation of static interoperability



Not enough to validate temporal interoperability



Behavior modeling



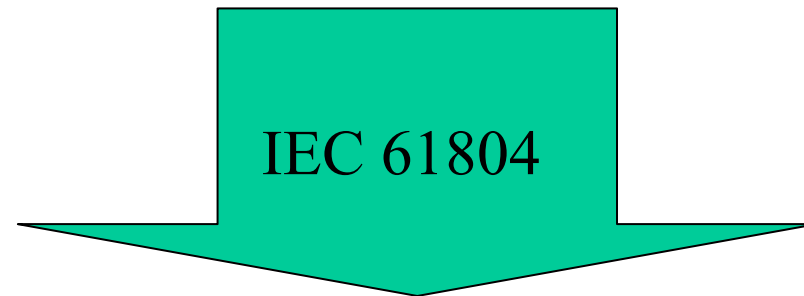
Application simulation

- **Description of the standard**
- **Adaptation taking into account the behavior**
- **Simulation of the application**
- **Example**
- **Conclusion**

- **Description of the standard**
- **Adaptation taking into account the behavior**
- **Simulation of the application**
- **Example**
- **Conclusion**

Function block concept

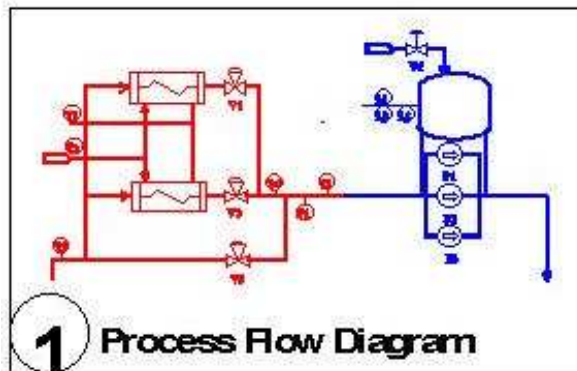
Goal: Have a common standard by which the users can be assured of compatible, interworkable, interconnectable, interoperable and interchangeable of the device they choose



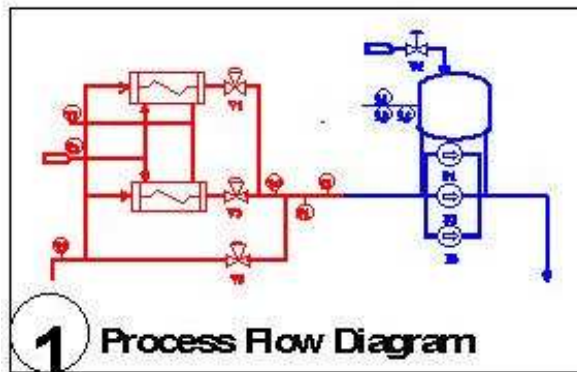
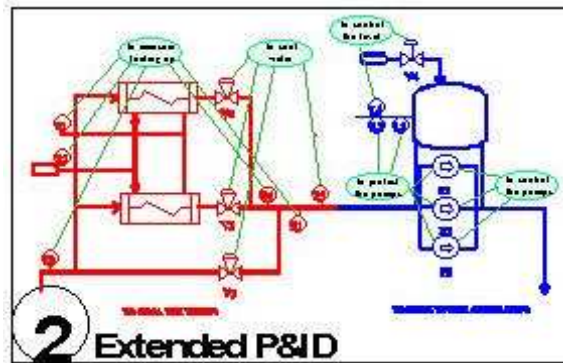
« function block is an encapsulation of data and algorithms to provide a specific function, which can be self understanding »



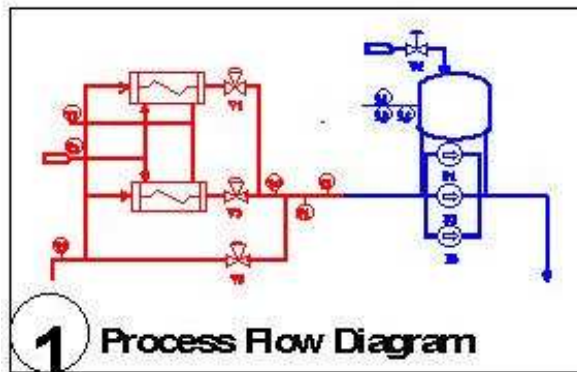
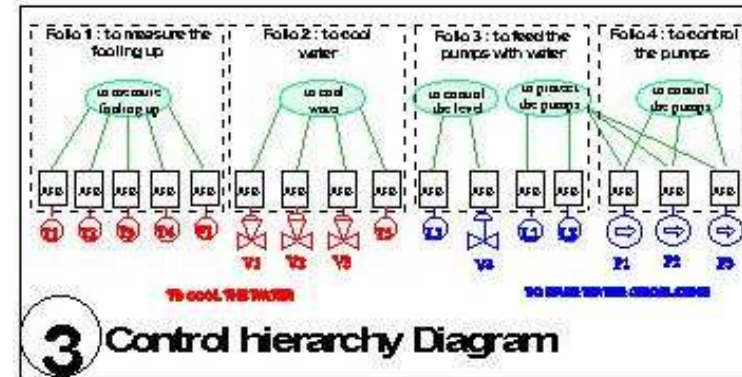
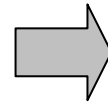
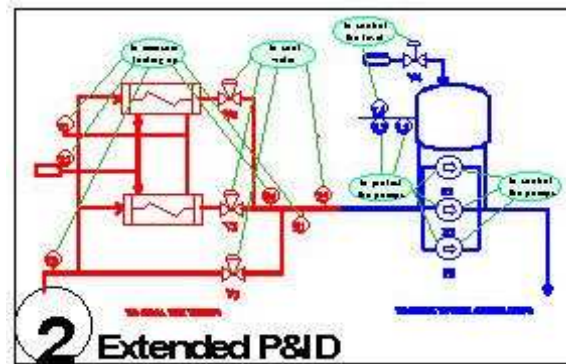
Function Blocks – How to use



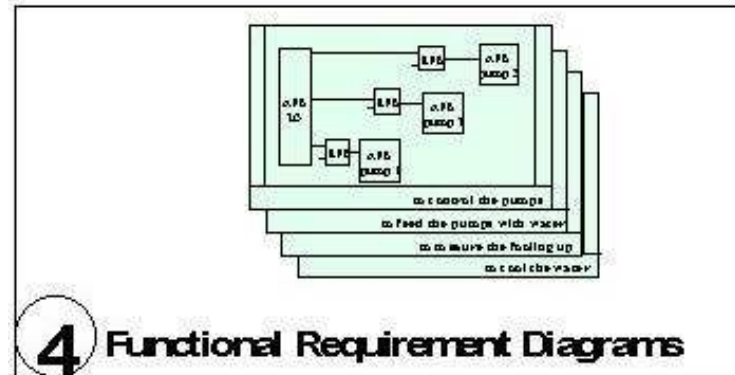
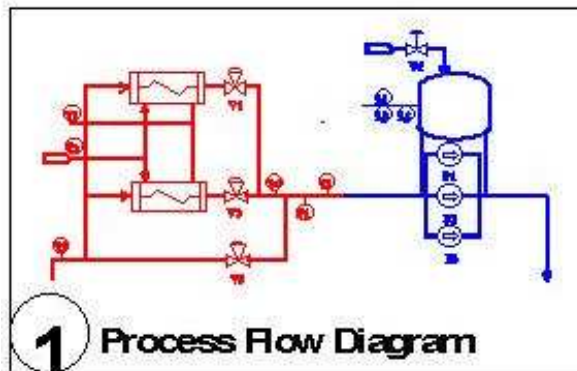
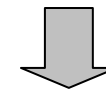
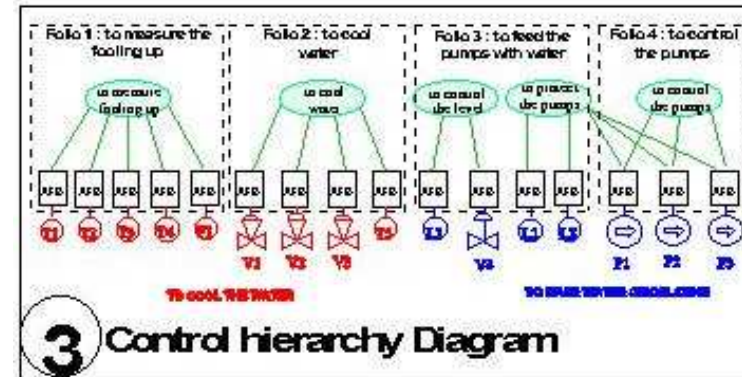
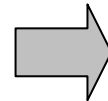
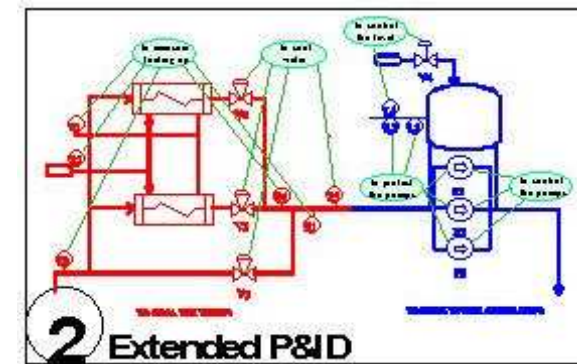
Function Blocks – How to use

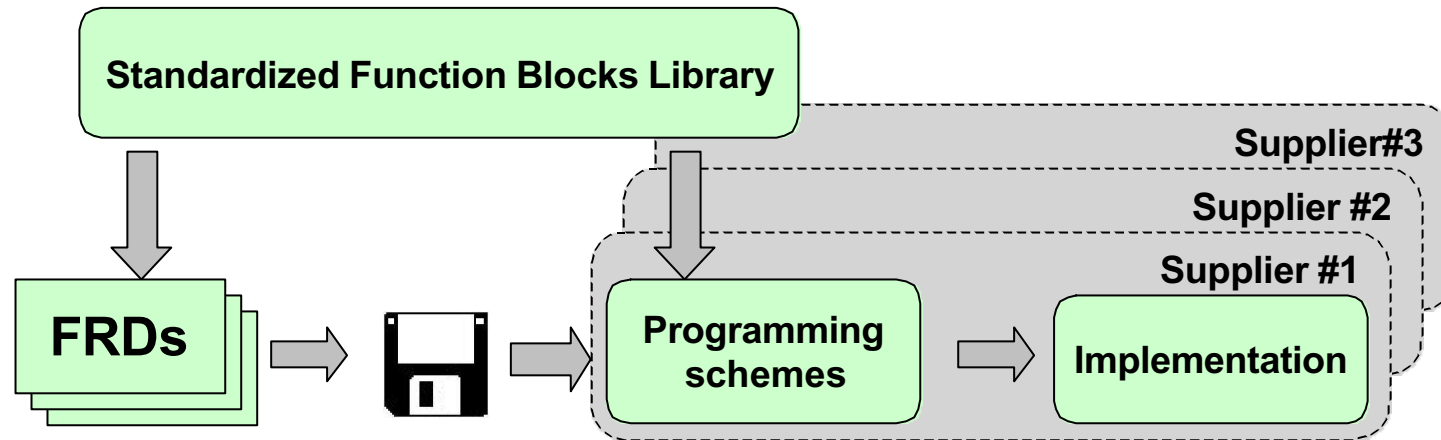


Function Blocks – How to use



Function Blocks – How to use





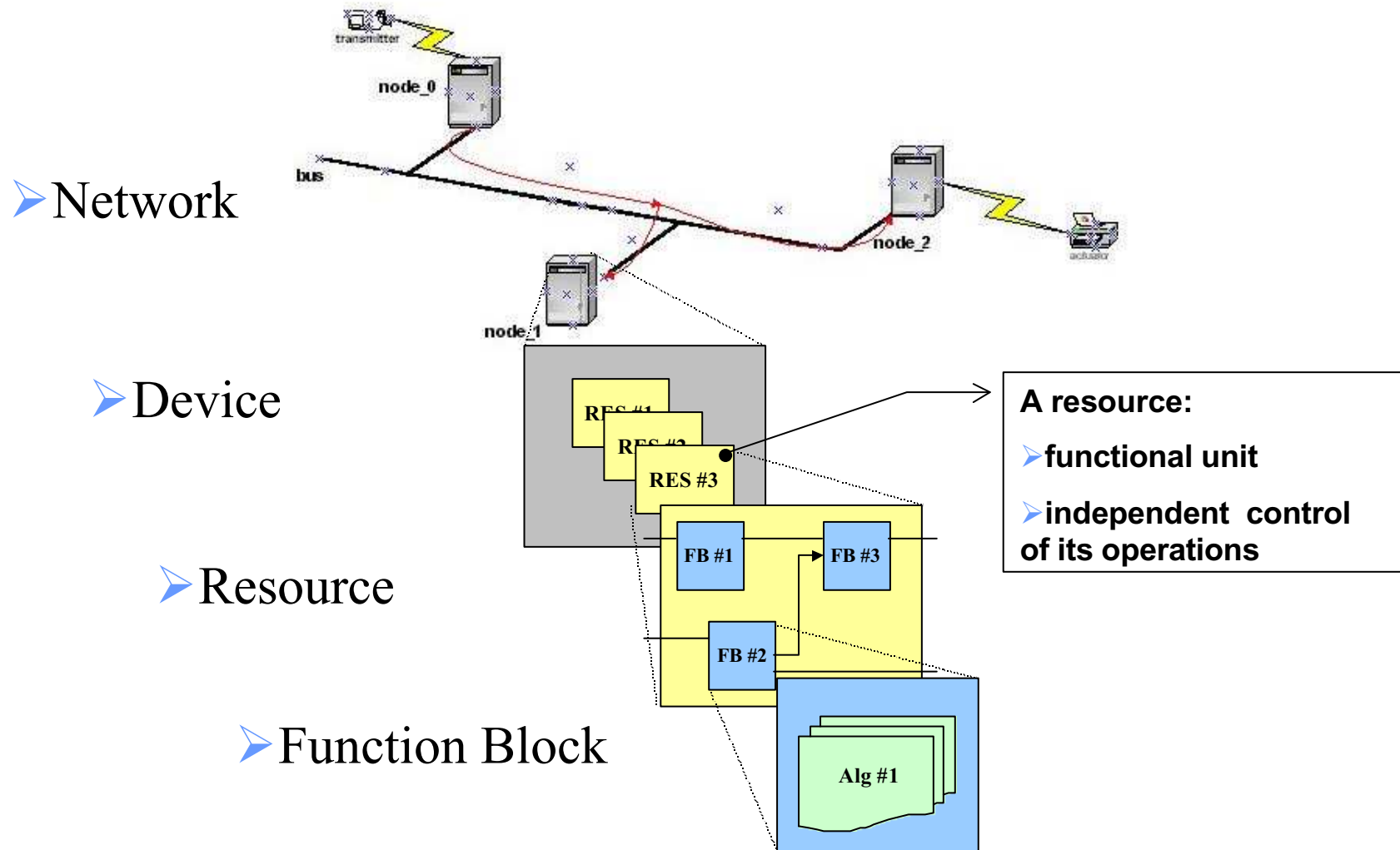
Objectives of FRDs is to describe:

- Control functions
- Performance
- Constraints

Problem:

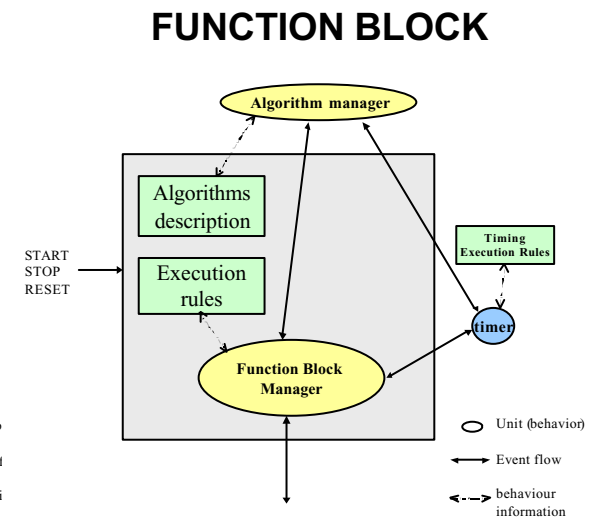
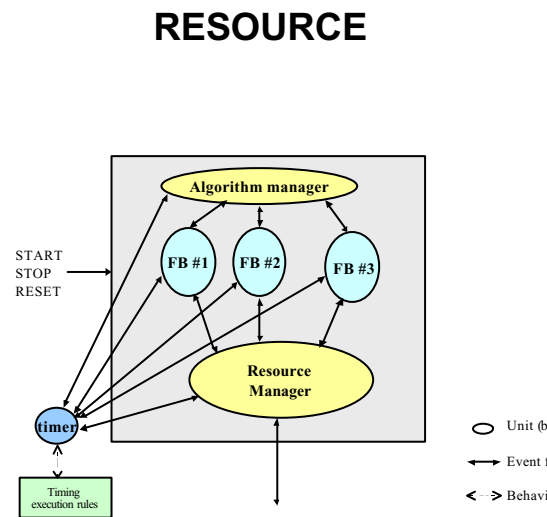
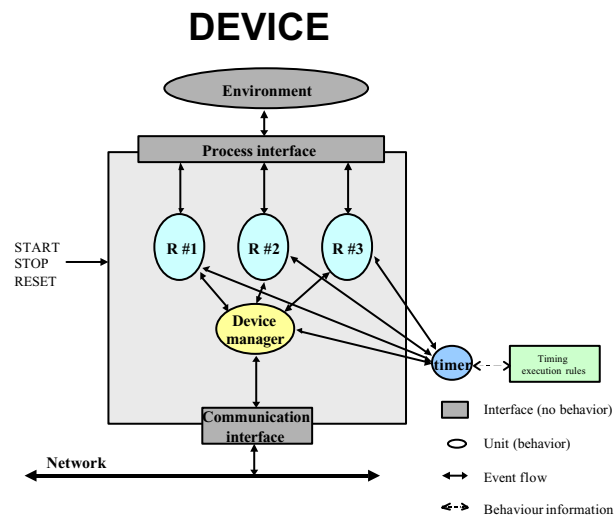
- FRDs consider only static aspects
- How to describe temporal characteristics

Considered Architecture model



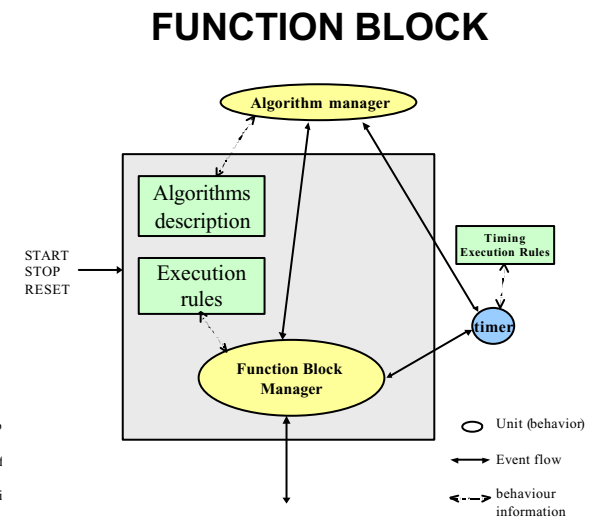
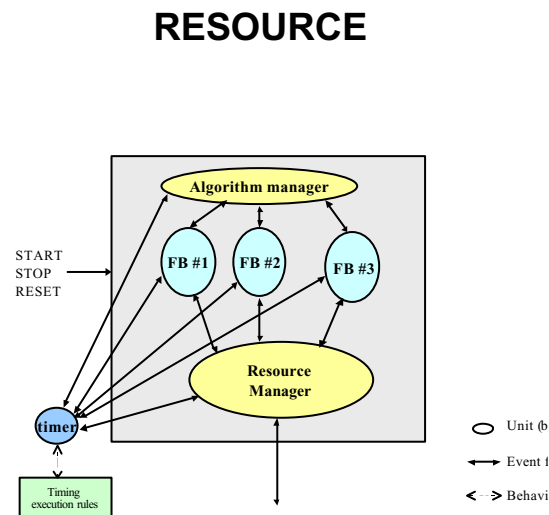
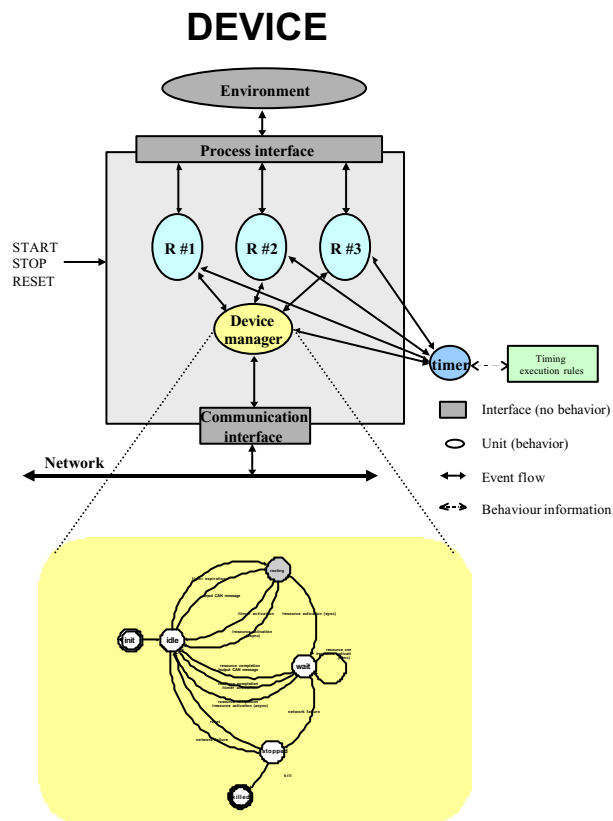
- Description of the standard
- **Adaptation taking into account the behavior**
- Simulation of the application
- Example
- Conclusion

Architecture model



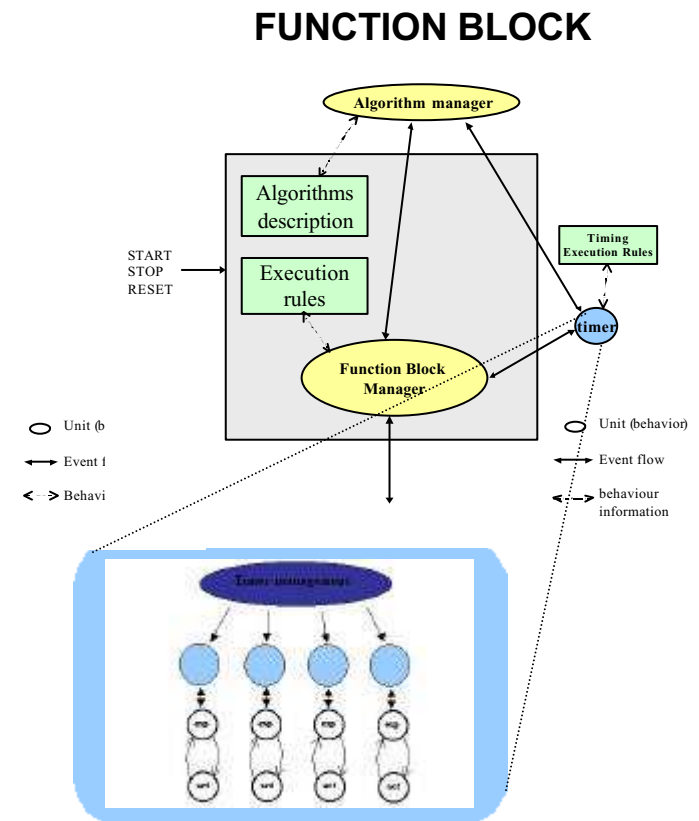
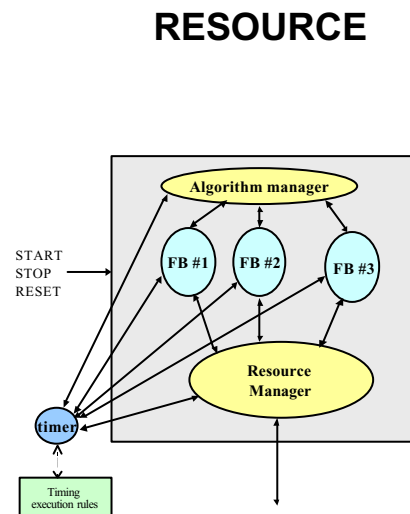
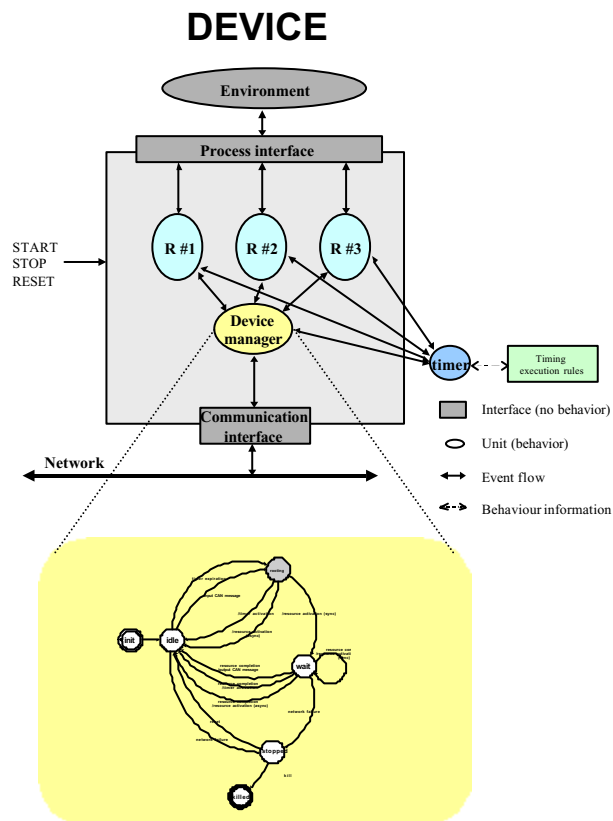
Architecture model

- For each level : introduction of a unit to manage the behavior;



Architecture model

- For each level : introduction of a unit to manage the behavior;
- Introduction of a global timer to modelize real timer and synchronization management.

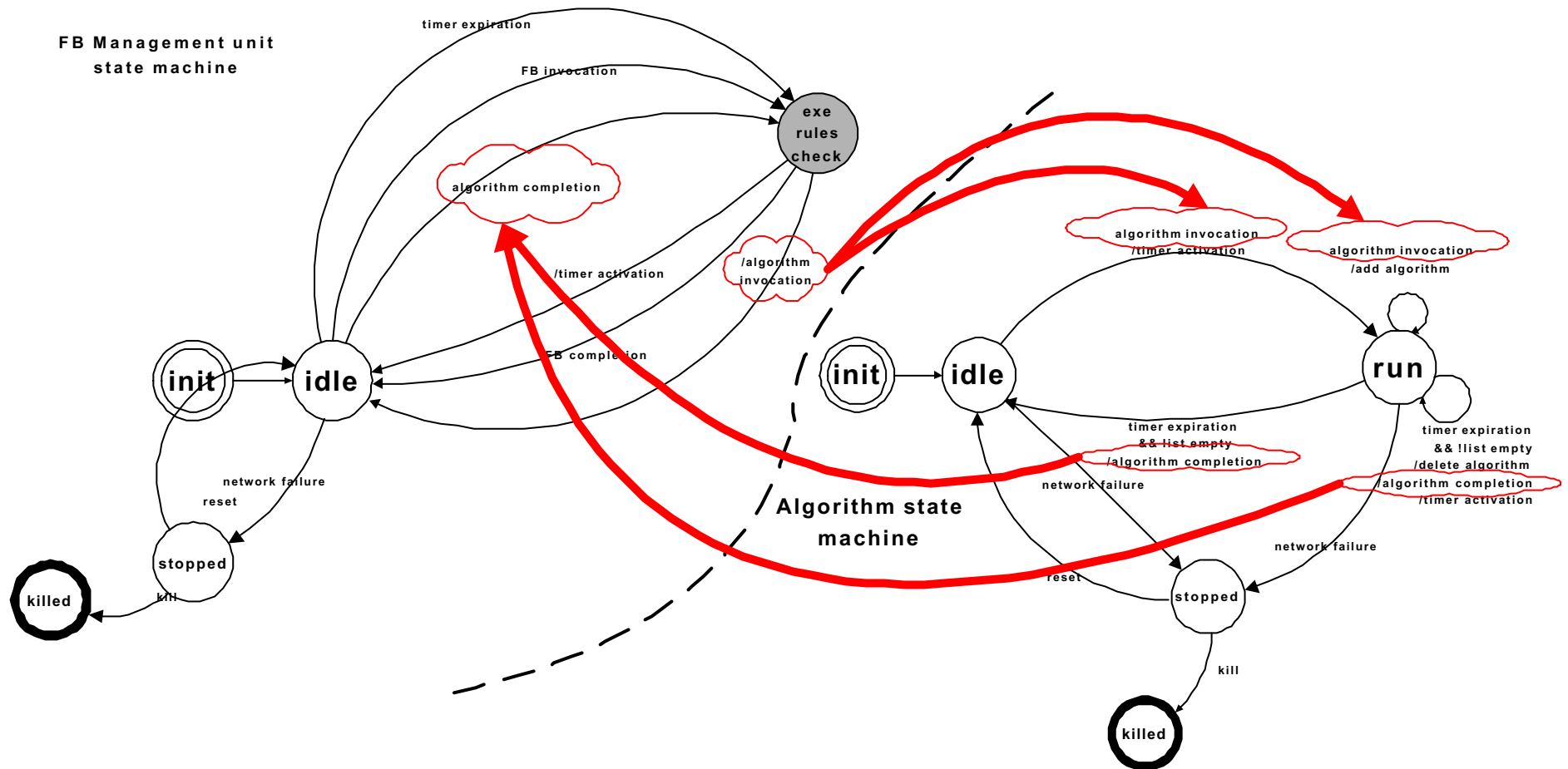


Execution model

- We only consider events (*necessary for the execution scheduling*)
 - Synchronous/asynchronous
 - Cyclic/acyclic

Execution model

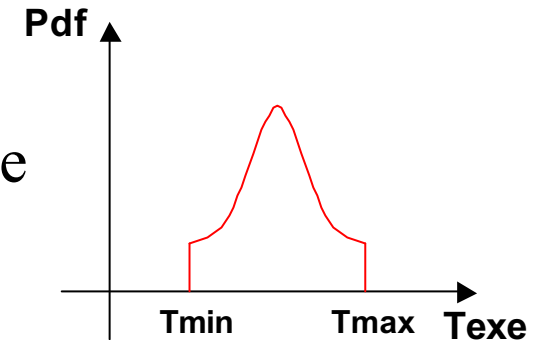
- We only consider events (*necessary for the execution scheduling*)
 - Synchronous/asynchronous
 - Cyclic/acyclic



Application behavior

Algorithm description

- Algo#ID: algorithm identifier
- [MinTime,MaxTime]: bounds of execution time
- Probability density function: distribution



Execution rules

« ON *event* IF *condition* DO *action* »

- Action to perform when an event occurs
- Condition: predicate on local variables
- Action: operations on local variables / event sending

« ON FB_invocation(num_port) DO Send (Algo_invocation(algoID), channelID) »

- Extensions

« AT *time* IF *condition* DO *action* »

« EACH *time* IF *condition* DO *action* »

Plan

- Description of the standard
- Adaptation taking into account the behavior
- **Simulation of the application**
- Example
- Conclusion

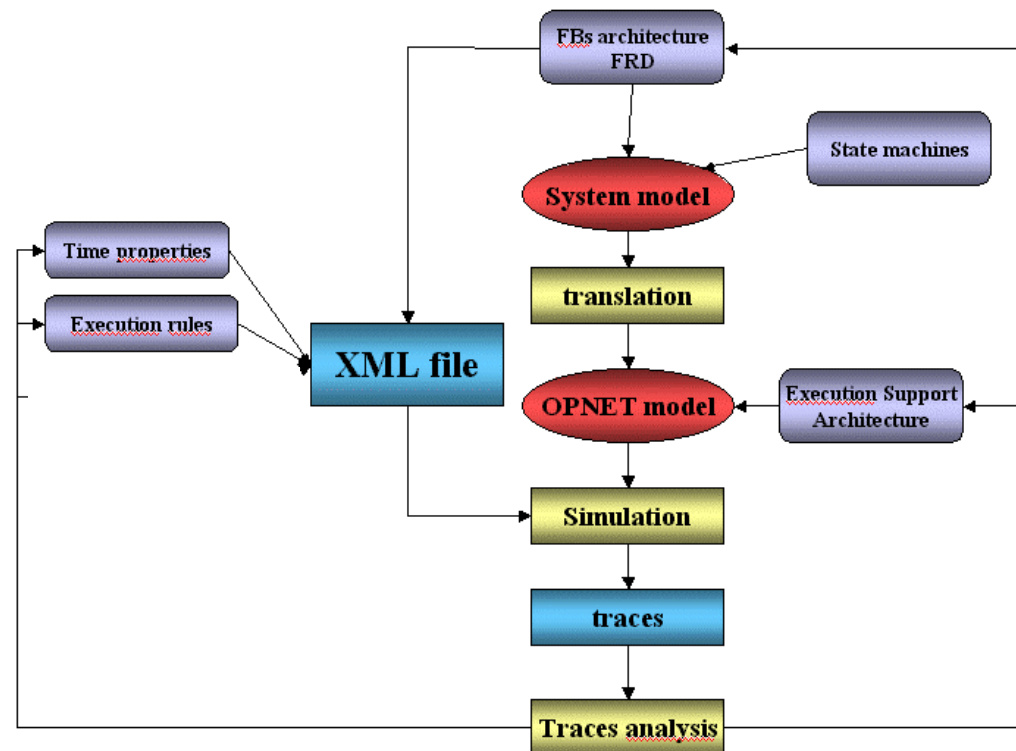
Simulation

Goals

- Validating system temporal properties
- System dimensioning regard to execution support architecture
- Distribution over resources

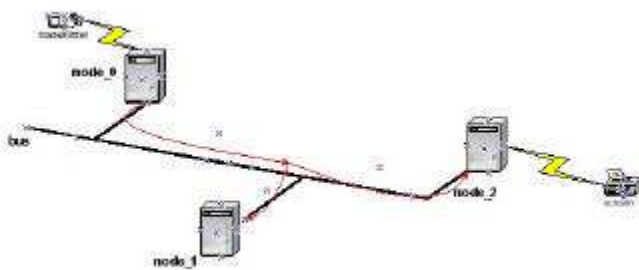
OPNET tool

- performance evaluation
 - Communication networks
 - distributed systems

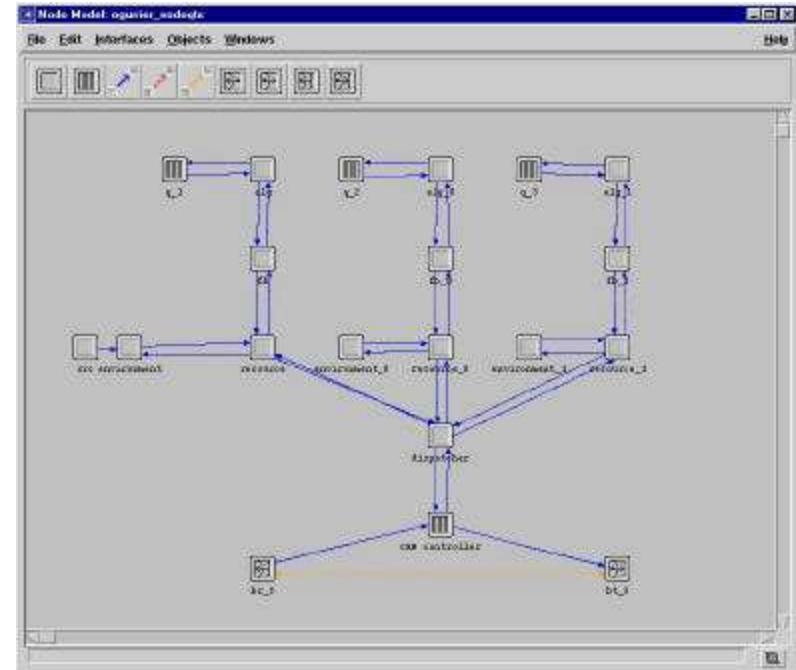


- Description of the standard
- Adaptation taking into account the behavior
- Simulation of the application
- **Example**
- Conclusion

Example



- Network: CAN protocol
- Device: composed by three resources
- Transmitters: random messages



- Delay between transmitter and actuator
- Average values

Delay increases in time



System is not well dimensioned

Conclusion

Behavior model for Function blocks

- Rely on the structural architecture
- Dynamic behavior of an application = execution rules
- Execution rules used by state machines

Simulation

- Validating system temporal properties
- System dimensioning
- Distribution over resources

Future works

- Simulate complex applications
- Extend to other system features
- Detail the description of the behavior (*operational modes*)
- Time constraints in the execution rules (*timed automata*)